Weakly-supervised approaches for sound event detection

Thomas Pellegrini
colab. Léo Cances, Patrice Guyot

16-04-2019

Outline

LUDAU

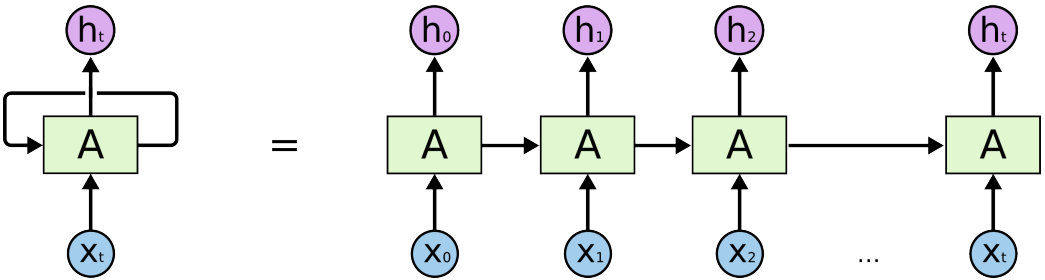
ANR JCJC Project (ANR-18-CE23-0005-01)

- Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning
- Start: 01 Oct. 2018
- End: 31 March 2022 (duration: 42 months)
- PhD student: Léo Cances, Post-doc to be hired

Outline

Generic RNNs

Unrolled RNN



RNN:

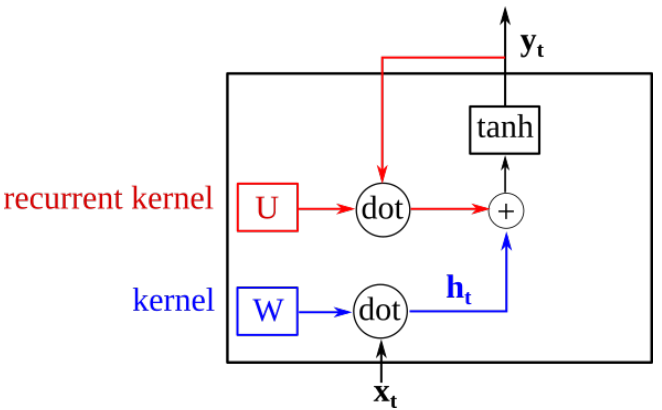
- a hidden state $\mathbf{h}$
- an optional output $\mathbf{y}$ which operates on a variable-length sequence $\mathbf{x} = (x_1, \ldots, x_T)$

At each time step t, $\mathbf{h}_t$ is updated by:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, x_t)$$

where f is a non-linear function such as a sigmoid or an LSTM, or a GRU, etc

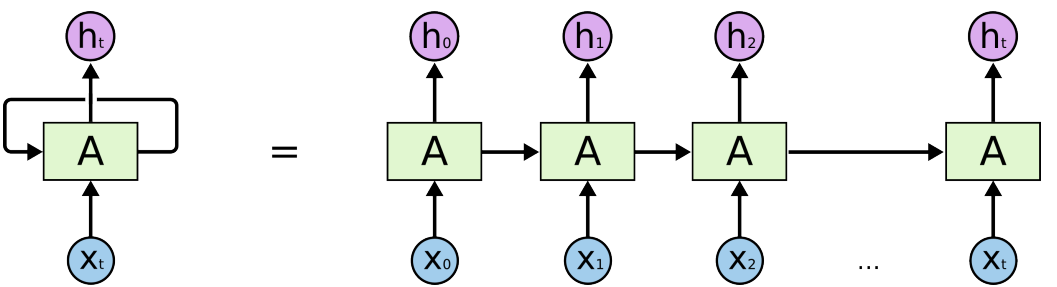
Simple RNN (keras version)



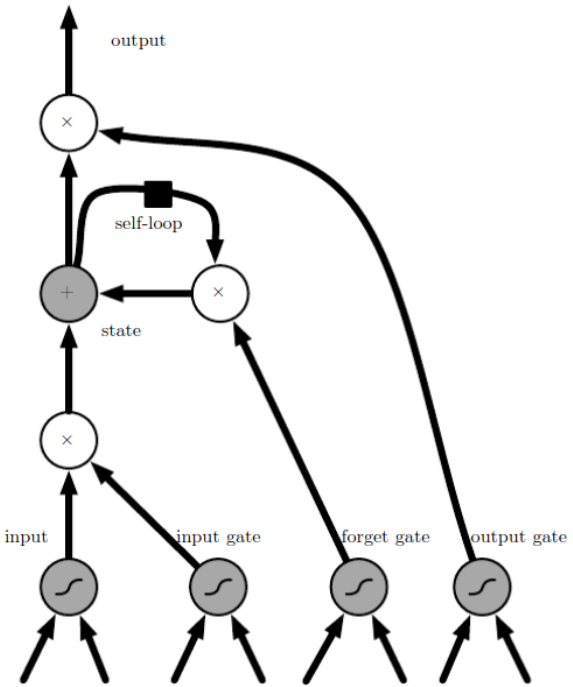
$$\mathbf{h_t} = \mathbf{x_t} \cdot W$$
$$\mathbf{y_t} = \tanh(\mathbf{h_t} + \mathbf{y_{t-1}} \cdot U)$$

Example: 10 cells, $\mathbf{x_t} \in \mathbb{R}^d \rightarrow W \in \mathbb{R}^{d \times 10}, U \in \mathbb{R}^{10 \times 10}, \mathbf{h_t}, \mathbf{y_t} \in \mathbb{R}^{10}$

Unrolled RNN



Long Short Term Memory (LSTM)



[Hochreiter and Schmidhuber, 1997], [Gers, 2000]
Image from `https://www.deeplearningbook.org/contents/rnn.html`

Gated Recurrent Units (GRU)

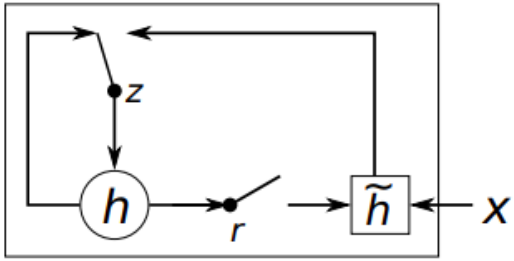


Figure 2: An illustration of the proposed hidden activation function. The update gate z selects whether the hidden state is to be updated with a new hidden state $\tilde{h}$. The reset gate r decides whether the previous hidden state is ignored. See

- Two gating units instead of three in LSTM
- shown to exhibit better performance on smaller datasets

[Cho et al, 2014], [Chung et al, 2014]

Gated Recurrent Units (GRU)

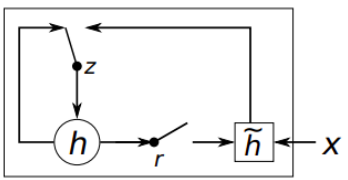
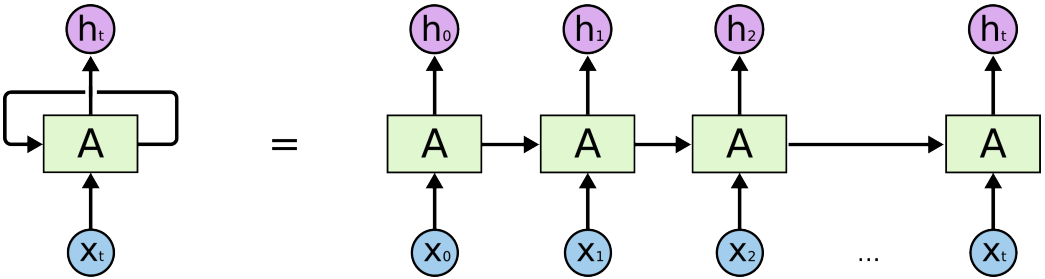


Figure 2: An illustration of the proposed hidden activation function. The update gate z selects whether the hidden state is to be updated with a new hidden state $\tilde{h}$. The reset gate r decides whether the previous hidden state is ignored. See

- Hidden-state update candidate at time t:
  $\tilde{h} = \tanh(W_h[\Gamma_r \odot h^{t-1}, x^t] + b_h)$
- Actual hidden-state output:
  $h^t = \Gamma_z \odot \tilde{h} + (1 - \Gamma_z) \odot h^{t-1}$
- Gates:
  - Reset gate:
    $\Gamma_r = \sigma(W_r[h^{t-1}, x^t] + b_r)$
  - Update gate:
    $\Gamma_z = \sigma(W_z[h^{t-1}, x^t] + b_z)$

Generic RNNs

Unrolled RNN



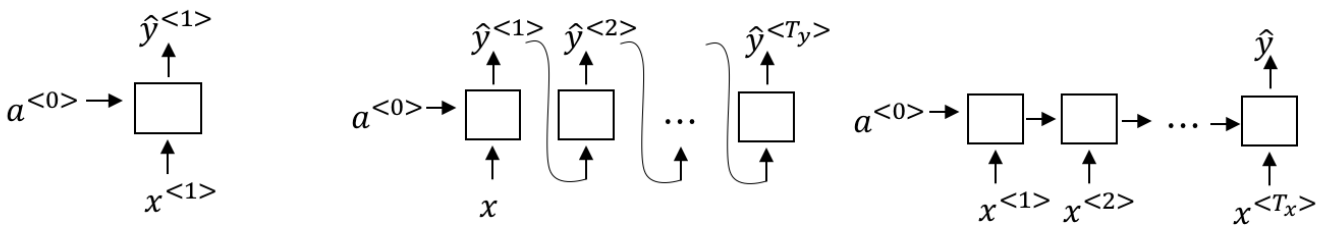
- An RNN can be trained to predict the next symbol in a sequence
- In that case, the output at each timestep t is the conditional distribution $p(x_t|x_{t-1}, \ldots, x_1)$
- For language modeling for instance, we would use a softmax function:

$$p(x_{t,j}|x_{t-1}, \ldots, x_1) = \frac{\exp(\mathbf{w_j h_t})}{\sum_{i=1}^{K} \exp(\mathbf{w_i h_t})}$$

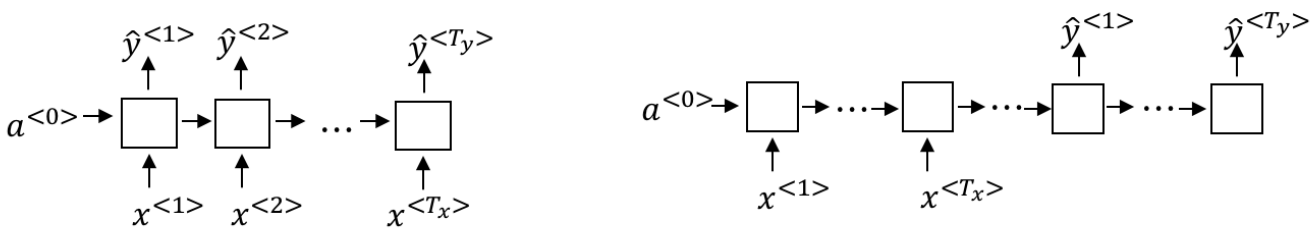
  for all possible words $j = 1 \ldots K$, where $\mathbf{w_j}$ are the rows of a weight matrix W
- By combining these probabilities, we can compute the probability of the sequence $\mathbf{x}$ using

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t|x_{t-1}, \ldots, x_1)$$

Summary of RNN types



One to one | One to many | Many to one

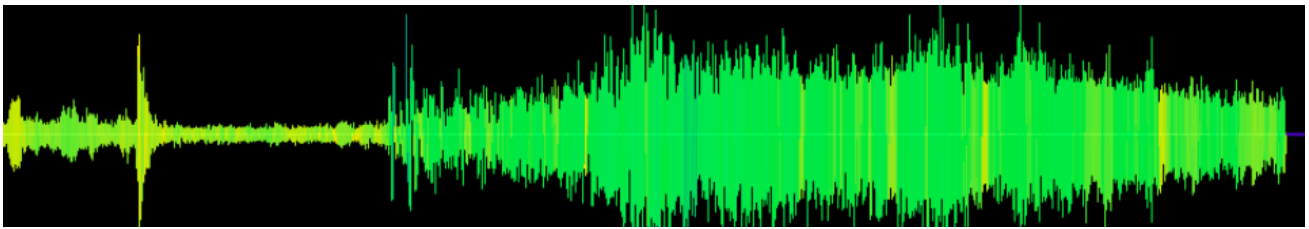


Many to many | Many to many

Andrew Ng

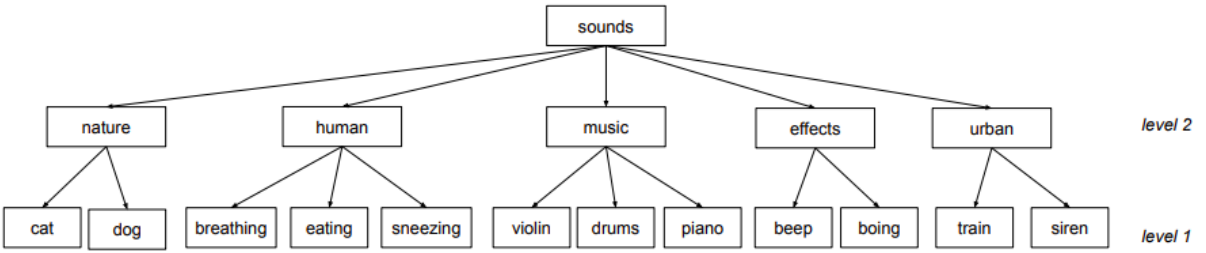
Outline

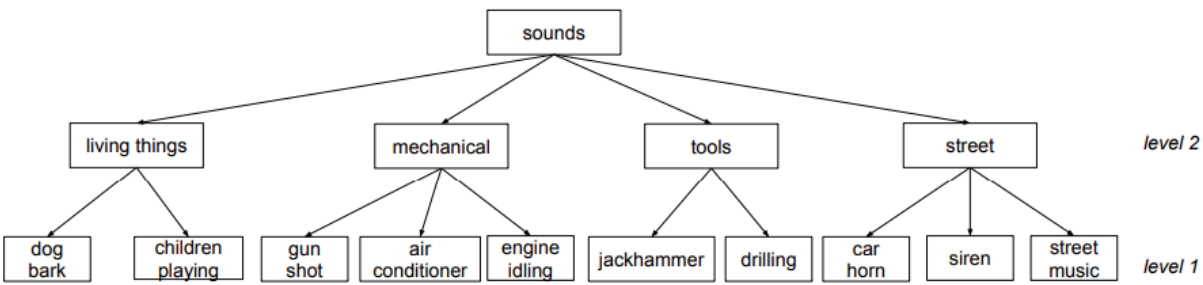
Machine listening



... extracting information from sound

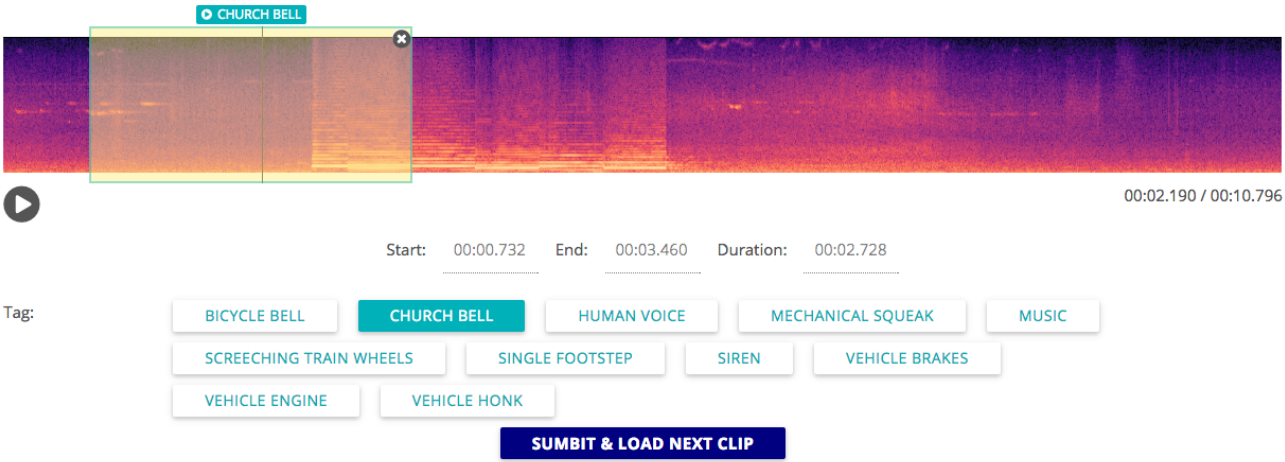
Machine listening





... making sense of sound

Why weakly-supervised learning?



`https://github.com/CrowdCurio/audio-annotator/`

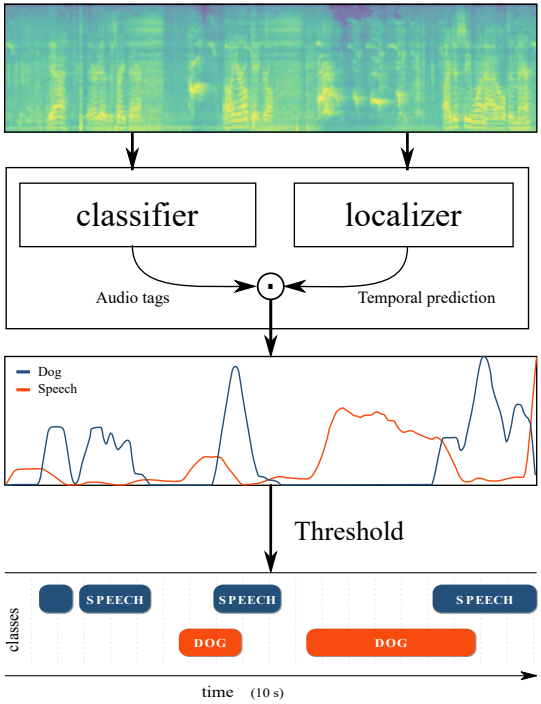
M. Cartwright, A. Seals, J. Salamon, A. Williams, S. Mikloska, D. MacConnell, E. Law, J. Bello, and O. Nov. "Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations." In Proceedings of the ACM on Human-Computer Interaction, 1(1), 2017.

Why weakly-supervised learning?

Issues regarding data annotation

- What to annotate? Which granularity?
- Annotations are limited, expensive and time-consuming
- Annotations may introduce biases
- Annotations are noisy, "label-noise"
- "Digital labor", les travailleurs du clic, Antonio Casilli

Sound Event Detection



- Multi-label classification problem
- *Weak labels*: audio tags, $\mathbf{y} \in \{0,1\}^C$: $\rightarrow$ many-to-one RNN
- *Strong labels*: frame-level tags, $Y \in \{0,1\}^{T \times C}$: $\rightarrow$ many-to-many RNN

Weakly-labeled SED: problem statement

- We want to perform:
  - "audio tagging": infer *weak* labels, $\hat{\mathbf{y}}$
  - "localization" or SED: infer *strong* labels, $\hat{Y}$
- Input: a sequence of T.F. representation vectors, $X = \{\mathbf{x_1}, \ldots \mathbf{x_T}\}$
- **Only weak labels are available for learning**
- In a recording weakly labeled as *Dog*, some acoustic frames will comprise dog barking and others will not

DCASE 2018 task 4 dataset

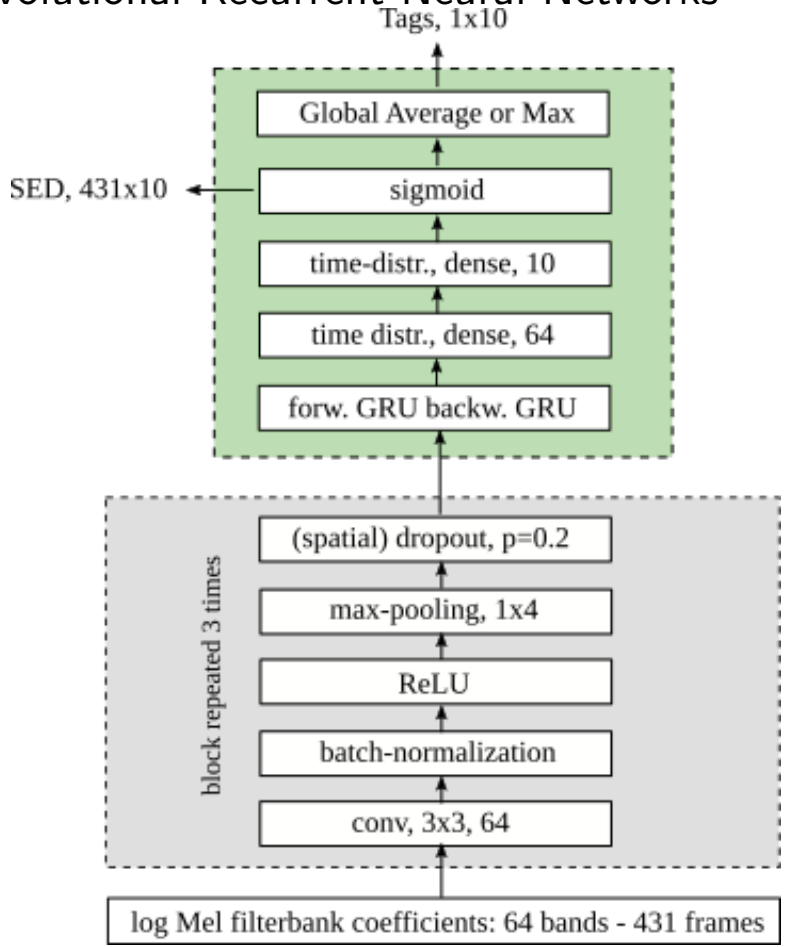
- A subset of Audioset
- 10-second audio clips
- 10 classes
- Labeled in-domain data
- Train: 4.4 h, 1578 files, 2244 occurrences, 20% for dev
- Evaluation: 0.8 h, 288 files with strong labels

Weakly-labeled SED: approaches

Three main approaches:

1. Baseline brute-force approach: "False strong labeling"
2. Attention mechanisms: Gated Linear Units, for ex.
3. MIL-inspired approach: max MIL pooling function and variants

Models: Convolutional Recurrent Neural Networks



Loss function for classification: cross-entropy

"False strong labeling" (FSL): strong labels == weak label

$$\mathrm{loss}(\{X^k, y_c^k\}) = \mathrm{binCE}(y_c^k, \hat{y}_{tc}^k)$$

$$\mathrm{binCE}(y_c^k, \hat{y}_{tc}^k) = -y_c^k \log \hat{y}_{tc}^k - (1 - y_c^k) \log(1 - \hat{y}_{tc}^k)$$

Attention-based approaches

1. Weighted-Gated Recurrent Units (w-GRU)
2. Gated Linear Units (GLU)
3. Other attention mechanisms

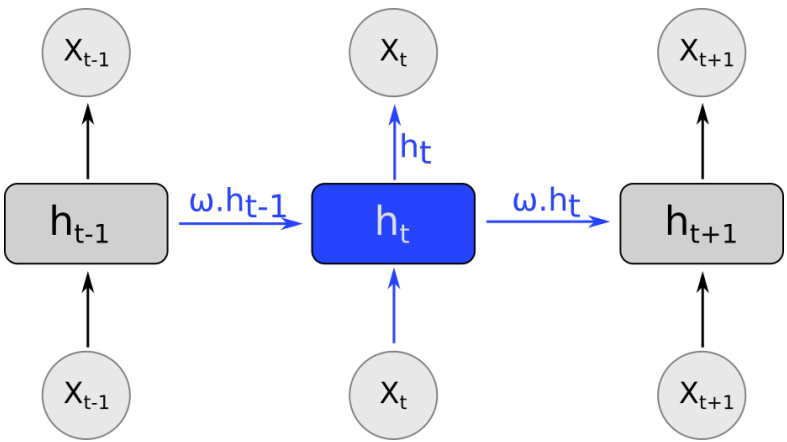
Weighted-GRU

At training time:

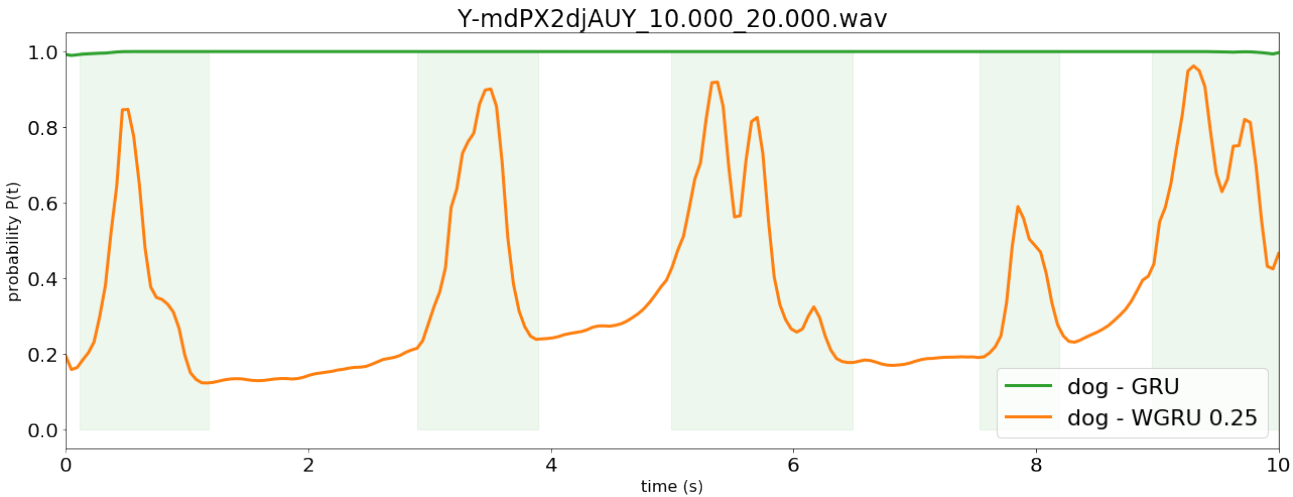
$$h_t = g(Wx_t + Uh_{t-1} + b)$$

At inference time:

$$h_t = g(Wx_t + U\omega h_{t-1} + b)$$



Weighted-GRU



Limitations: ad-hoc approach, does not work with all sound types

SOUND EVENT DETECTION FROM WEAK ANNOTATIONS: WEIGHTED GRU VERSUS MULTI-INSTANCE LEARNING

Léo Cances, Thomas Pellegrini, Patrice Guyot

Gated Linear Units (GLU)

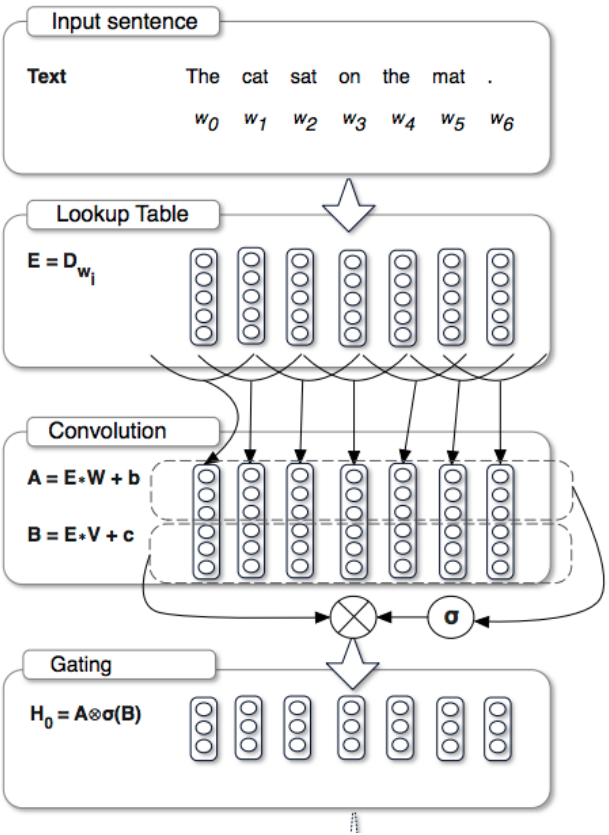
Language Modeling with Gated Convolutional Networks

Yann N. Dauphin [1] Angela Fan [1] Michael Auli [1] David Grangier [1]

- Pre-dominant approach to language modeling: recurrent neural networks
- GLU: non-recurrent alternative with stacked convolutions, allow parallelization over sequential tokens

Gated Linear Units (GLU)

$$\mathbf{h_l}(X) = (X * W + \mathbf{b}) \odot \sigma(X * V + \mathbf{c})$$

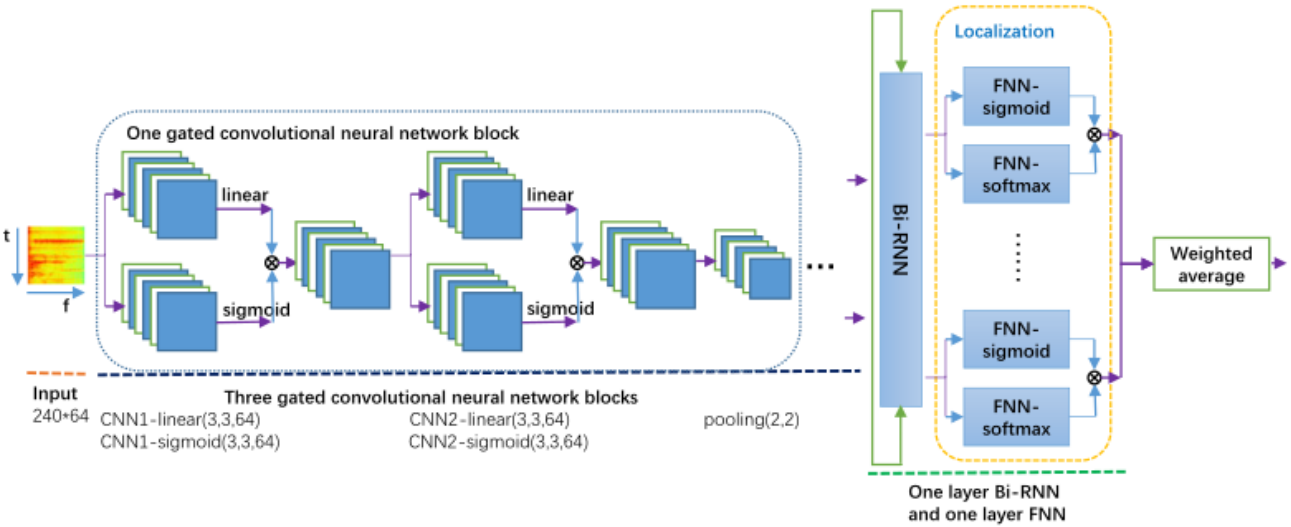


- $X \in \mathbb{R}^{N \times m}$ input of layer l
- learnable: $W, V \in \mathbb{R}^{k \times m \times n}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$
- Similar to LSTM, the $\sigma(X * V + \mathbf{c})$ gates control the information pass
- The gates replace standard activation functions (such as ReLU)
- Linear activation to help gradients flow
- Input shift to prevent using information from future words
- GLUs wrapped in a residual block
- Output activation function: adaptive Softmax [Grave et al, 2016]

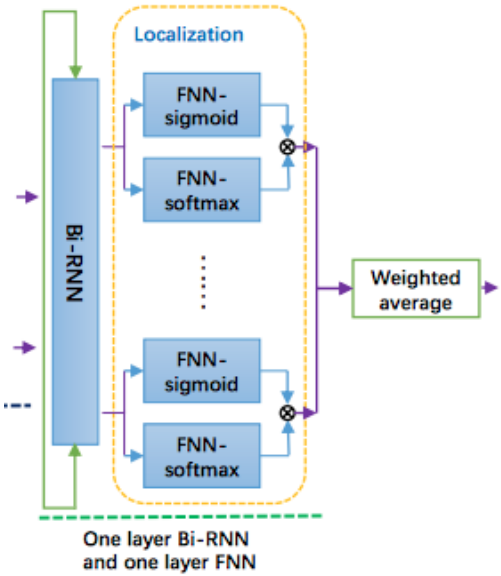
Gated CNN in SED

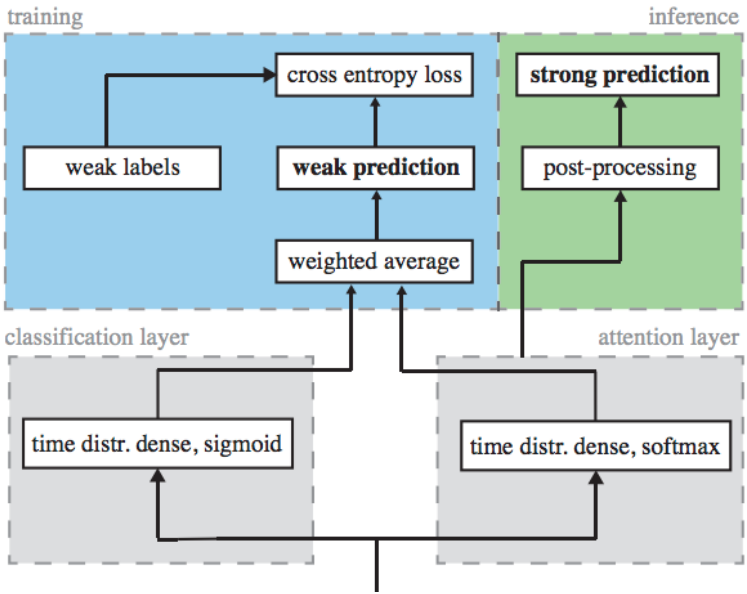
LARGE-SCALE WEAKLY SUPERVISED AUDIO CLASSIFICATION USING GATED CONVOLUTIONAL NEURAL NETWORK

Yong Xu, Qiuqiang Kong*, Wenwu Wang, Mark D. Plumbley*



Gated CNN in SED





Picture from [Robert Harb, Franz Pernkopf, DCASE-2018]

Final audio tag prediction:

$$y_c = \frac{\sum_t^T \mathbf{z}_c^{\mathrm{cla}}(t) \odot \mathbf{z}_c^{\mathrm{att}}(t)}{\sum_t^T \mathbf{z}_c^{\mathrm{att}}(t)}$$

Multiple Instance Learning

- MIL terminology
  - $X = \{\mathbf{x_1}, \ldots, \mathbf{x_T}\}$: a bag or a set
  - $\mathbf{x_i}$: an instance
  - No dependency nor ordering among instances
  - A single binary label Y
  - Unknown individual labels: $y = \{y_1, \ldots, y_T\}$

MIL assumptions:

$$Y = \begin{cases} 0, & \text{iff } \sum_t y_t = 0, \\ 1, & \text{otherwise} \end{cases}$$

In a compact form:

$$Y = \max_t \{y_t\}$$

MIL: how to model the bag-level prob $P(Y = 1|X)$?

- MIL assumption: $P(Y = 1|X)$ must be **permutation invariant** since no ordering nor dependency between instances $\mathbf{x_t}$
- Instances: "monomials"

Theorem 1. *A scoring function for a set of instances X, $S(X) \in \mathbb{R}$, is a symmetric function (i.e., permutation-invariant to the elements in X), if and only if it can be decomposed in the following form:*

$$S(X) = g\Big(\sum_{\mathbf{x} \in X} f(\mathbf{x})\Big), \tag{3}$$

where f and g are suitable transformations.

Theorem 2. *For any $\varepsilon > 0$, a Hausdorff continuous symmetric function $S(X) \in \mathbb{R}$ can be arbitrarily approximated by a function in the form $g\big(\max_{\mathbf{x} \in X} f(\mathbf{x})\big)$, where $\max$ is the element-wise vector maximum operator and f and g are continuous functions, that is:*

$$|S(X) - g\big(\max_{\mathbf{x} \in X} f(\mathbf{x})\big)| < \varepsilon. \tag{4}$$

MIL: how to model $P(Y = 1|X)$?

Instance-level approach

- f: instance-level classifier that returns scores for each instance
- σ: aggregates the scores
- g: Identity function

Bag-level (embedding-level) approach

- f: maps instances to embeddings
- σ: maps the emb. to a single bag emb.
- g: bag-level classifier
- σ: "MIL pooling", basic non-learnable choices:
  - $\mathbf{z} = \max_{t=1\ldots T}\{\mathbf{h_t}\}$
  - $\mathbf{z} = 1/T \sum_{t=1}^{T}\{\mathbf{h_t}\}$

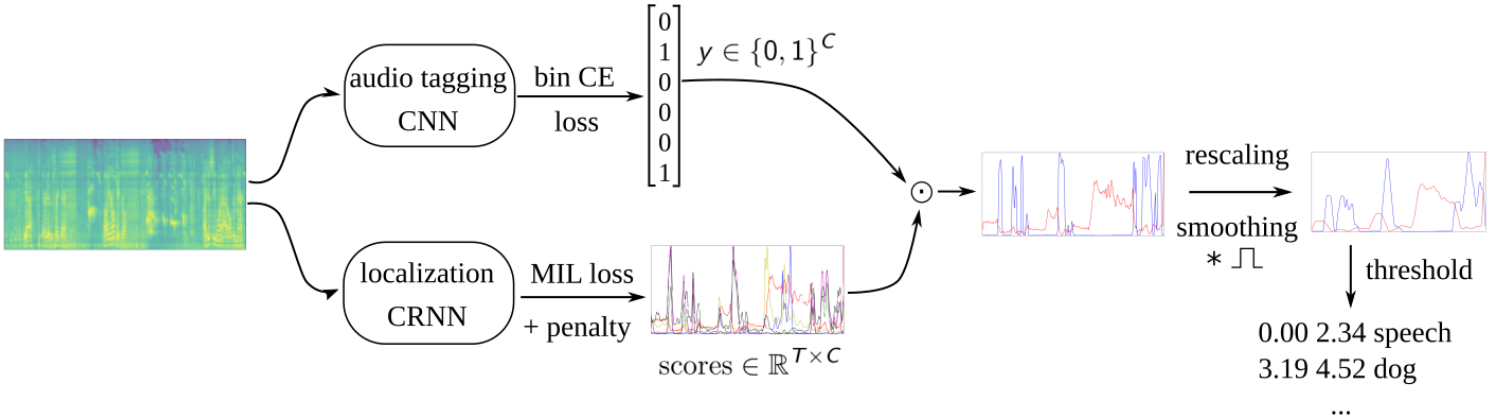
MIL loss for WL-SED

$$\mathrm{loss}(\{X, y_c\}) = \mathrm{binCE}(y_c, \max_t \hat{\mathbf{y}}_{\mathbf{tc}}) \tag{1}$$

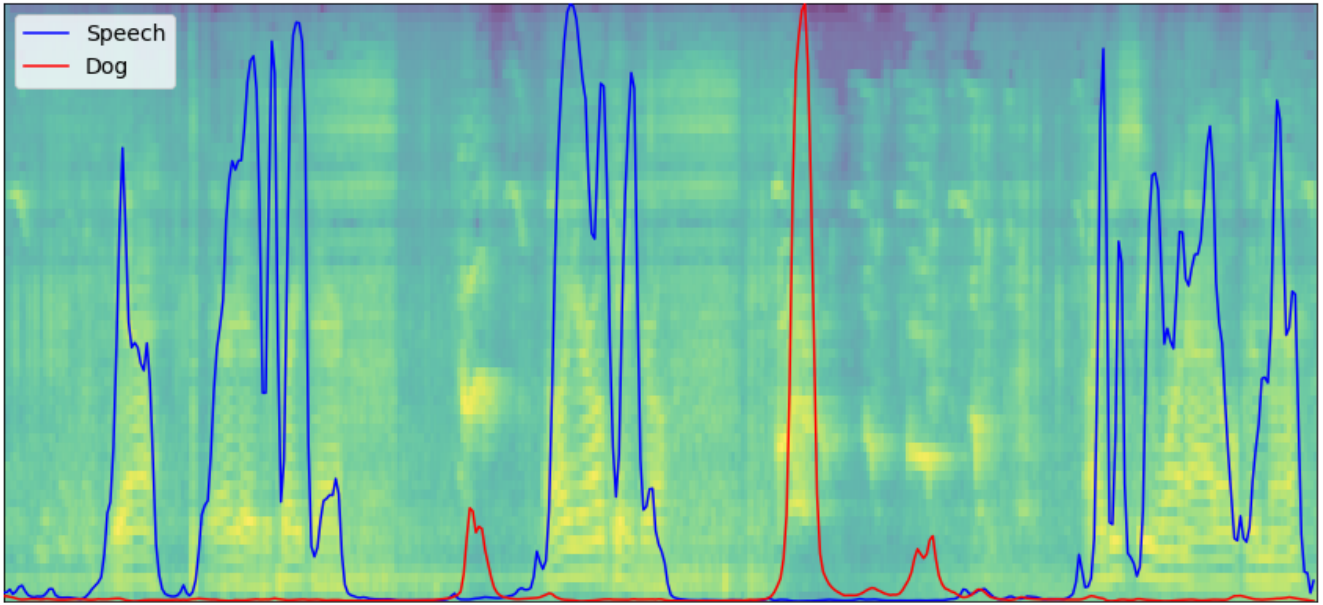
where

- $y_c \in \{0, 1\}$: ground truth label for class c
- $\hat{\mathbf{y}}_{\mathbf{tc}} \in [0, 1]^T$: temporal predictions for class c

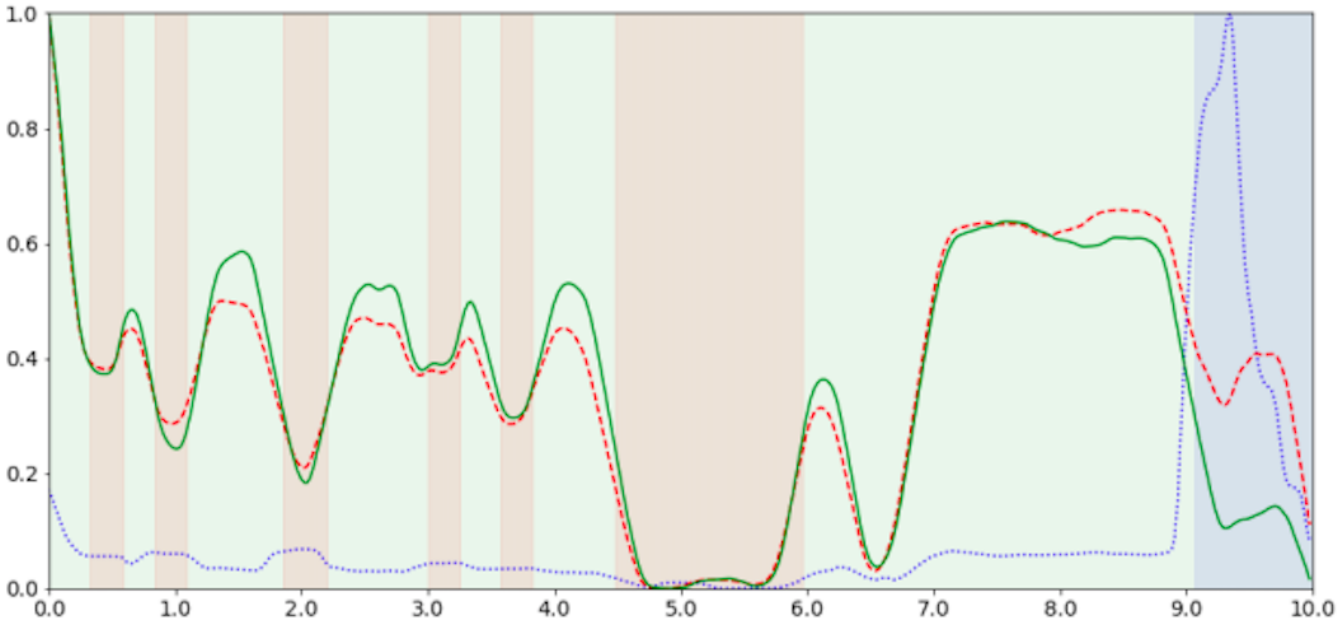
System overview



Example OK

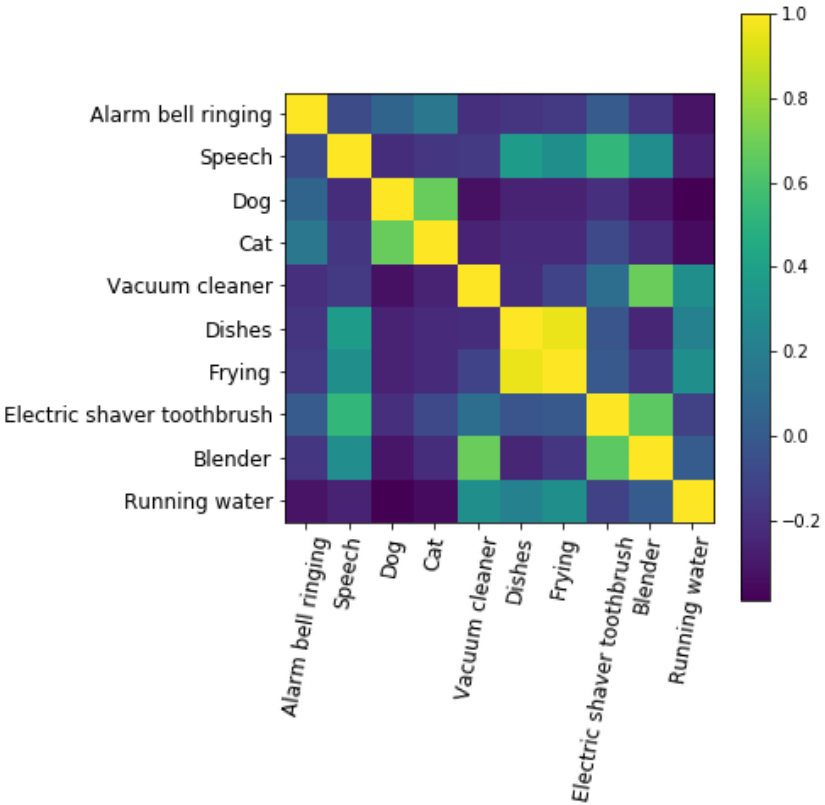


Example KO



- Issue in distinguishing some classes

Prediction correlations



- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying*??

Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying*??

→ A caveat of the MIL pooling function (here max)

- When a class most often co-occurs with another one
- When max predictions are scarce (a single frame possibly): during training the gradient flows on those frames only

Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying*??

→ A caveat of the MIL pooling function (here max)

- When a class most often co-occurs with another one
- When max predictions are scarce (a single frame possibly): during training the gradient flows on those frames only

Solutions?

- Choose another MIL pooling function (ongoing work)
- Penalize for prediction similarity (joint work with Léo Cances, `arxiv.org/abs/1901.03146`)

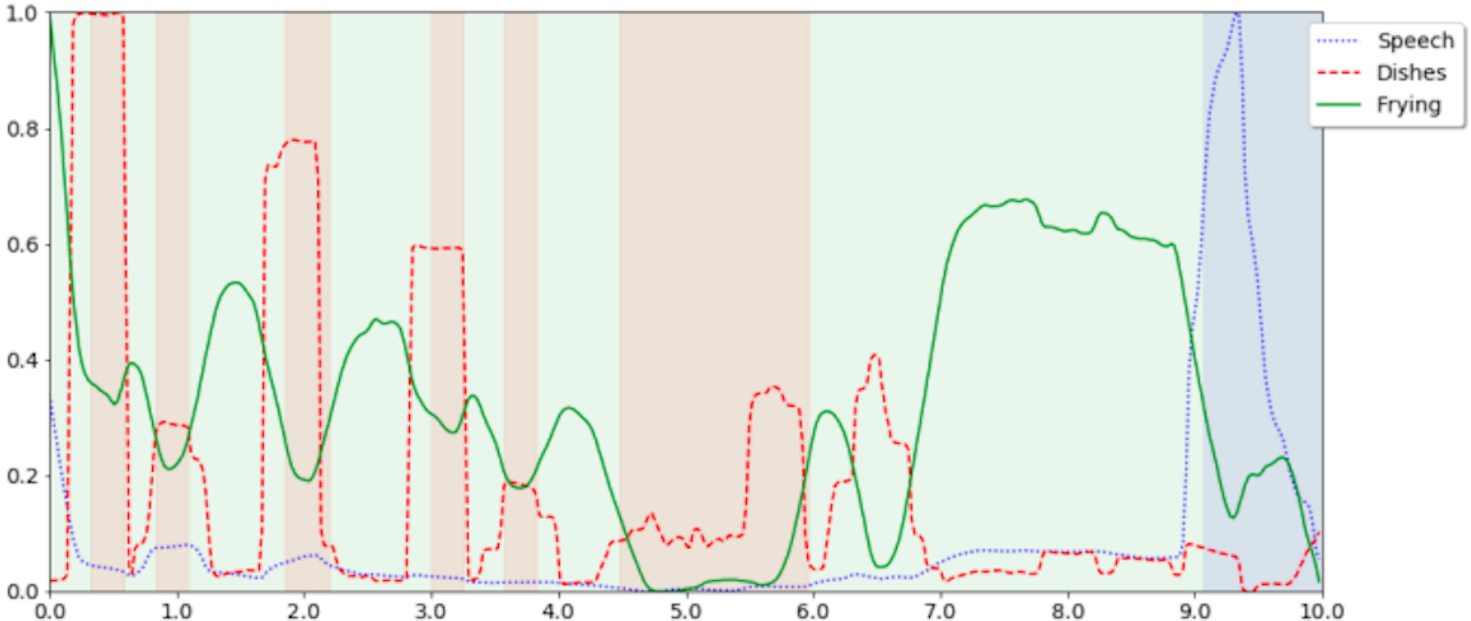
Adding a similarity penalty

$$\begin{aligned}\mathrm{loss}(\{X, y_c\}) &= \mathrm{binCE}(y_c, \max_t \hat{\mathbf{y}}_{\mathbf{tc}}) \\ &\quad + \alpha\, y_c \sum_{l \neq c}^{C} y_l \max(0, \cos(\hat{\mathbf{y}}_{:\mathbf{l}}, \hat{\mathbf{y}}_{:\mathbf{c}}))\end{aligned}$$

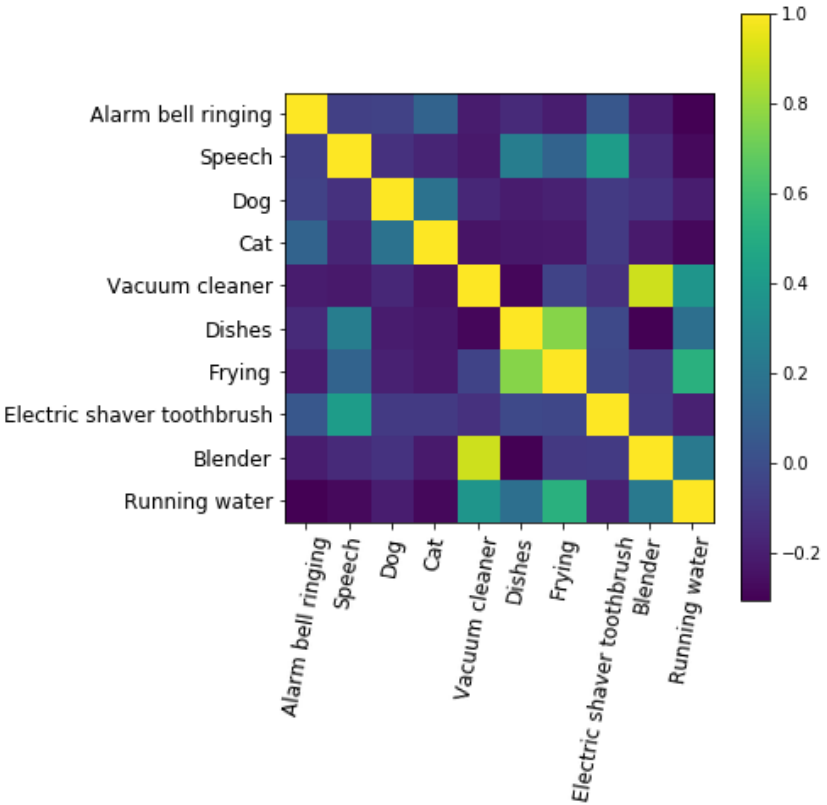
Results on Eval 2018

Approach	Official Baseline	MIL	GLU-MIL	GLU-MIL+cos	JiaKai	Liu
F-score (%)	10.8	18.86	22.60	26.20	32.4	29.9
Alarm / bell / ringing	4.8	30.1	29.0	30.4	49.9	46.0
Blender	12.7	28.8	23.3	27.7	38.2	27.1
Cat	2.9	22.8	28.7	30.3	3.6	20.3
Dishes	0.4	1.0	0.0	19.0	3.2	13.0
Dog	2.4	20.1	19.8	20.9	18.1	26.5
Electric shaver / toothbrush	20.0	7.7	6.2	19.1	48.7	37.6
Frying	24.5	0.0	27.6	21.2	35.4	10.9
Running water	10.1	17.9	13.3	13.2	31.2	23.9
Speech	0.1	36.7	37.6	35.0	46.8	43.1
Vacuum cleaner	30.2	23.5	40.6	45.2	48.3	50.0

Adding a similarity penalty

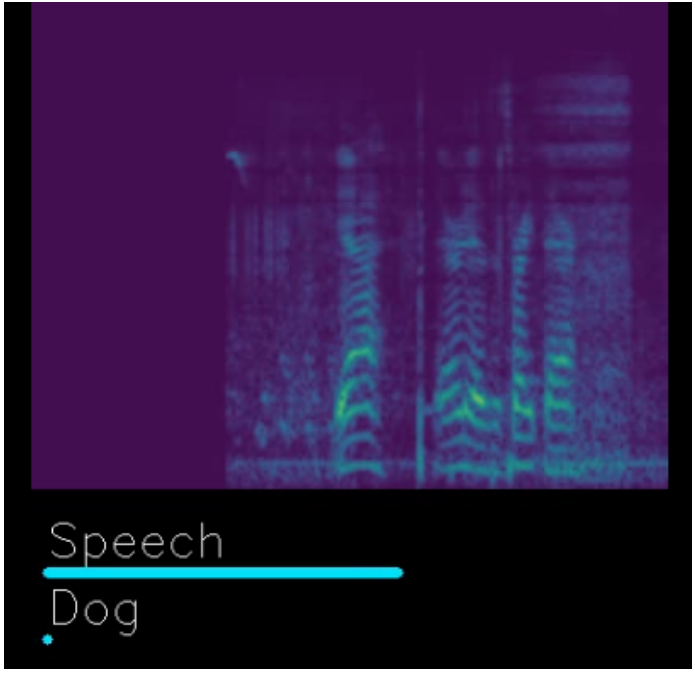


Prediction correlations



- Less confusions!

demo



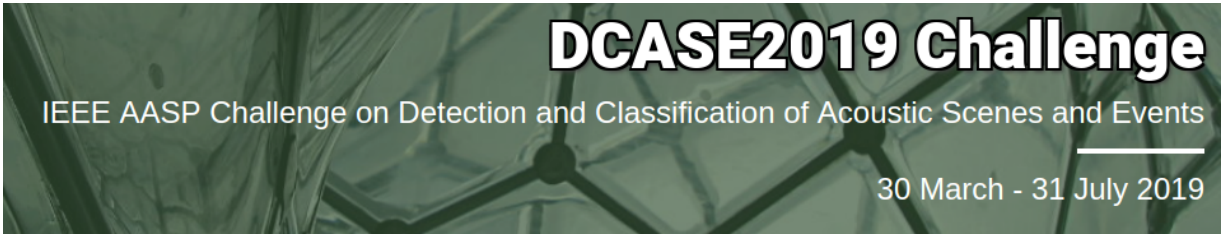
Discussion on the cos penalty

Pros

- Increased the class-discriminative power of the networks

Open questions

- Larger variance
- Tends to force events to not overlap (but OK for the challenge)
- Better to penalize the prediction distributions? e.g. a Kullback-Leibler penalty?
- Worked for SED, generalization to other MIL problems?



Tasks

- Acoustic Scene Classification
- Audio tagging with noisy labels and minimal supervision
- Sound Event Localization and Detection
- Sound event detection in domestic environments
- Urban Sound Tagging

Task 4 focus:

- do we really need real but partially and weakly annotated data or is using simulated data sufficient? or do we need both?